

Article

A Novel Deep Learning Method Based on an Overlapping Time Window Strategy for Brain–Computer Interface-Based Stroke Rehabilitation

Lei Cao ^{1,†}, Hailiang Wu ^{1,†}, Shugeng Chen ^{2,†} , Yilin Dong ¹, Changming Zhu ¹, Jie Jia ^{2,*} and Chunjiang Fan ^{3,*}¹ Department of Artificial Intelligence, Shanghai Maritime University, Shanghai 201306, China² Department of Rehabilitation Medicine, Huashan Hospital, Fudan University, Shanghai 200040, China³ Department of Rehabilitation Medicine, Wuxi Rehabilitation Hospital, Wuxi 214001, China

* Correspondence: shannonjj@126.com (J.J.); fanchunjiang1980@163.com (C.F.);

Tel.: +86-139-1718-0962 (J.J.); +86-136-5619-0701 (C.F.)

† These authors contributed equally to this work.

Abstract: Globally, stroke is a leading cause of death and disability. The classification of motor intentions using brain activity is an important task in the rehabilitation of stroke patients using brain–computer interfaces (BCIs). This paper presents a new method for model training in EEG-based BCI rehabilitation by using overlapping time windows. For this aim, three different models, a convolutional neural network (CNN), graph isomorphism network (GIN), and long short-term memory (LSTM), are used for performing the classification task of motor attempt (MA). We conducted several experiments with different time window lengths, and the results showed that the deep learning approach based on overlapping time windows achieved improvements in classification accuracy, with the LSTM combined vote-counting strategy (VS) having achieved the highest average classification accuracy of 90.3% when the window size was 70. The results verified that the overlapping time window strategy is useful for increasing the efficiency of BCI rehabilitation.

Keywords: brain–computer interface; motor attempt (MA); EEG; deep learning method; overlapping time window



Citation: Cao, L.; Wu, H.; Chen, S.; Dong, Y.; Zhu, C.; Jia, J.; Fan, C. A Novel Deep Learning Method Based on an Overlapping Time Window Strategy for Brain–Computer Interface-Based Stroke Rehabilitation. *Brain Sci.* **2022**, *12*, 1502. <https://doi.org/10.3390/brainsci12111502>

Academic Editors: Konstantin V. Slavin and Gianluca Di Flumeri

Received: 7 September 2022

Accepted: 31 October 2022

Published: 5 November 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Stroke leads to high rates of disability and death worldwide [1]. To restore brain function affected by stroke, patients need to undergo rigorous rehabilitation. Currently, there are a variety of approaches to help restore motor function after a stroke, including the use of mirror therapy [2], virtual reality [3], aerobic exercise [4], and brain–computer interface (BCI) technology [5–7]. BCI technology can help patients recover independently and perform tasks by efficiently controlling additional devices, making it a good option for patients.

The use of a non-invasive BCI for motor rehabilitation has become a focus of current research and is considered a mainstream experimental method. Patients perform motor imagery (MI) or motor attempt (MA) tasks based on cues from the system. The BCI then decodes and converts the motor intents from the electroencephalogram (EEG) signals into commands and provides feedback according to the experimental protocol [8,9]. However, EEG signals may be unstable or random and show significant individual differences. In addition, EEG signals intended for the same behavior but collected at different times and under different circumstances may also have large differences; hence, it is difficult to classify EEG signals directly. Feature extraction and classification algorithms are needed to extract meaningful information from the multidimensional EEG signals [10].

Based on previous studies, the features extracted by traditional machine learning methods can be classified into three categories: spatial, time, and frequency. In dichotomous BCI tasks, the common spatial pattern (CSP) algorithm is the most common method for

extracting spatial features [11], as it is able to extract the spatially distributed components of each class from multichannel EEG signals. Many studies have expanded the CSP algorithm, and the filter bank CSP (FBCSP) developed from the CSP algorithm has achieved very good classification performance in MI-BCI [12]. Analyzing EEG signals in a time series can yield rich statistical features. Geethanjali et al. extracted seven time-domain features from EEG signals and classified them using linear discriminant analysis [13]. As many EEG signal features are reflected in the frequency domain, analysis of frequency domain features is important for BCIs. Furthermore, by converting EEG signals from the time domain to the frequency domain, the distribution and variation of EEG frequencies can be visualized. Chen et al. visualized event-related synchronization and event-related desynchronization in MA and MI tasks in different frequency bands [14]. Although the above method can be applied to MA-BCI and MI-BCI to some extent, they require prior knowledge and manually designed features combined with the use of machine learning for classification, which may present problems of insufficient feature extraction and low adaptability to different patients. Hence, many studies have tried to use deep learning (DL) to automatically learn features gathered from EEG signals for classification.

In contrast to traditional machine learning methods, DL does not require predefined feature vectors, as it can automatically learn latent and highly abstract features from raw EEG signals. The combination of BCI with DL methods has been used in the rehabilitation of patients. Lin et al. developed a convolutional neural network (CNN)-based model for predicting BCI rehabilitation outcomes [15]. Liang et al. used the long short-term memory (LSTM) neural network for generating motor trajectories of the lower-extremity exoskeleton for stroke rehabilitation [16] and a graph embedding-based model, Ego-CNN, for identifying key graph structures during MI [17]. However, BCI systems using DL methods require large amounts of EEG data for training models, which results in a bottleneck in therapy. At present, independent patients experience more difficulty performing control tasks due to tedious experimental steps, which often leads to less data collected and less than optimal accuracy for classification using DL methods.

For smaller datasets, data augmentation techniques have proven to be an effective way to improve the performance of DL models, and the approach is to generate more data from the original data for training the model. In previous studies, some have performed data enhancement by adding noise to the original EEG signals [18]. Sliding time windows are advantageous in augmenting EEG signal data. Hartmann et al. used an overlapping time window to expand a dataset of epileptic patients [19], and Zhang et al. extracted time and frequency domain features from multiple windows for a classification task of left versus right hand movements [20]. In addition, several studies have used generative adversarial network models for generating new data similar to the original EEG signal [21]. This approach can help to compensate for the inability to collect large amounts of EEG signals from patients during motor rehabilitation and further improve the performance of DL methods for classifying EEG signals.

The purpose of this paper was to improve the performance of DL on MA-BCI through data augmentation techniques to contribute to the rehabilitation training of patients. Specifically, we provide more accurate neurofeedback by improving the recognition accuracy of a patient's motor intention. To achieve this aim, we propose a DL method based on overlapping time windows for the classification tasks of MA-BCI. This study compares the classification performance of three different DL models on MA tasks. To investigate the effect of different time periods on BCI classification, we visualized the classification results on a time series and analyzed the differences in EEG signals at different time slices using the power spectral density topography of the brain.

2. Materials and Methods

The data used in our experiment were collected from 7 stroke subjects using BCI interventions. Demographic information and clinical data are reported in Table 1. All

subjects typically performed three sessions per week; one session included ninety trials, and each trial corresponded to one type of task: motor attempt (MA) or idle state (IS).

2.1. Experimental Protocol

Figure 1 shows the experimental protocol during rehabilitation training. The experimental setup consists of two components: a BCI module and a force feedback device. The BCI module is responsible for the collection and analysis of EEG signals, and the force feedback device is responsible for providing neurofeedback. The patient's stroke-affected hand was immobilized on the force feedback device that was controlled by the BCI system. When the experimental task was a motor attempt, patients continually attempted wrist extension with the affected hand. When the experimental task was in an idle state, they were ordered to rest and do nothing. The force feedback device drove the patient's stroke-affected hand to complete a wrist extension movement when the BCI system accurately identified the patient's motor intention. For incorrect identification, the device would stay stationary.

Table 1. Demographic information of the subjects.

Subject	Sex	Age	Affected Limb	Stroke Stage
Sub1	Male	31	Right	Subacute
Sub2	Male	40	Left	Subacute
Sub3	Male	42	Right	Subacute
Sub4	Male	47	Right	Subacute
Sub5	Male	36	Right	Subacute
Sub6	Male	30	Right	Subacute
Sub7	Male	65	Left	Subacute
Mean	-	41.6 ± 12.0	-	-

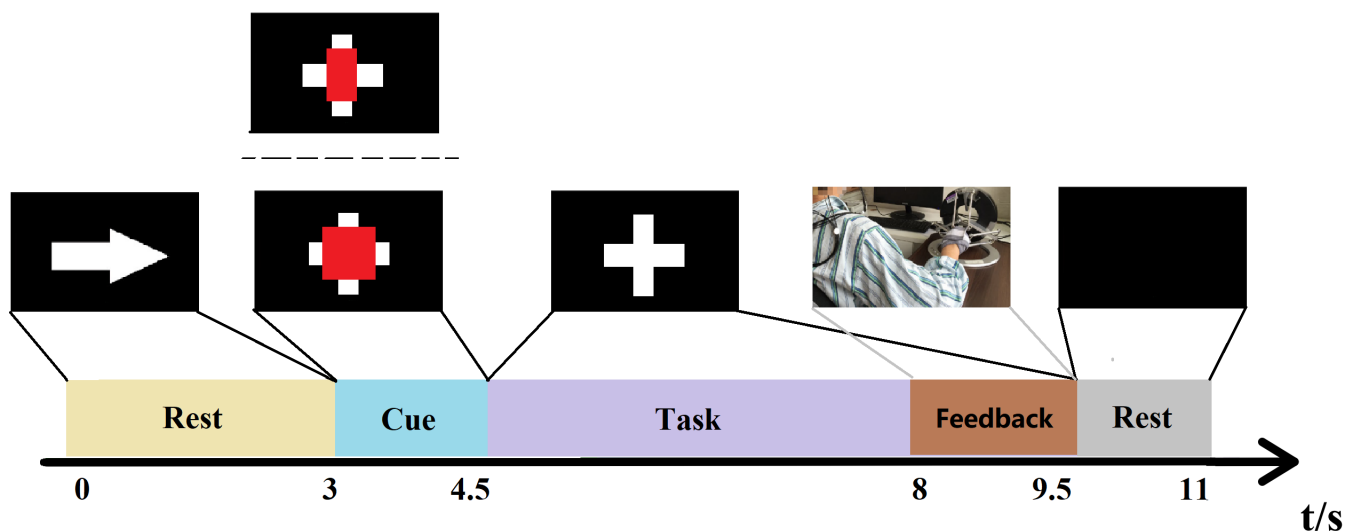


Figure 1. Experimental protocol of rehabilitation training. During the cueing period, a red rectangle is used to alert the user to perform specific tasks. When the cue is a red square, the patient will attempt wrist extension using the stroke-affected hand as hard as possible until the white cross disappears. When the cue is a red rectangle, the patient just needs to stay rested. The patient's stroke-affected hand was passively extended by the force feedback device when the system accurately identified the patient's motor intention.

2.2. Data Acquisition and Preprocessing

According to Figure 1, the EEG signals of each trial were recorded for 11 seconds and started with a white arrow image used to prompt the patient to be prepared. Three seconds later, a task cue (red geometrical shapes) was displayed on the screen, and the patient was

asked to perform either a movement attempt or a rest state. After the cue disappeared, the patient was told to continue performing the task following the cue until the white cross disappeared. Then, the patients rested for 1.5 s. The recorded signals were sampled by a 32-channel EEG cap, and the EEG electrodes were placed according to the international 10–20 system. The sampling frequency was 200 Hz. Data from 31 channels were used for calculation, and the filter range was 4 to 40 Hz. Some examples of the preprocessed EEG signals (C3, C4) from the motor function areas of the brain are shown in Figure 2. Five seconds of EEG before the white cross disappeared from each trial were extracted for training the model.

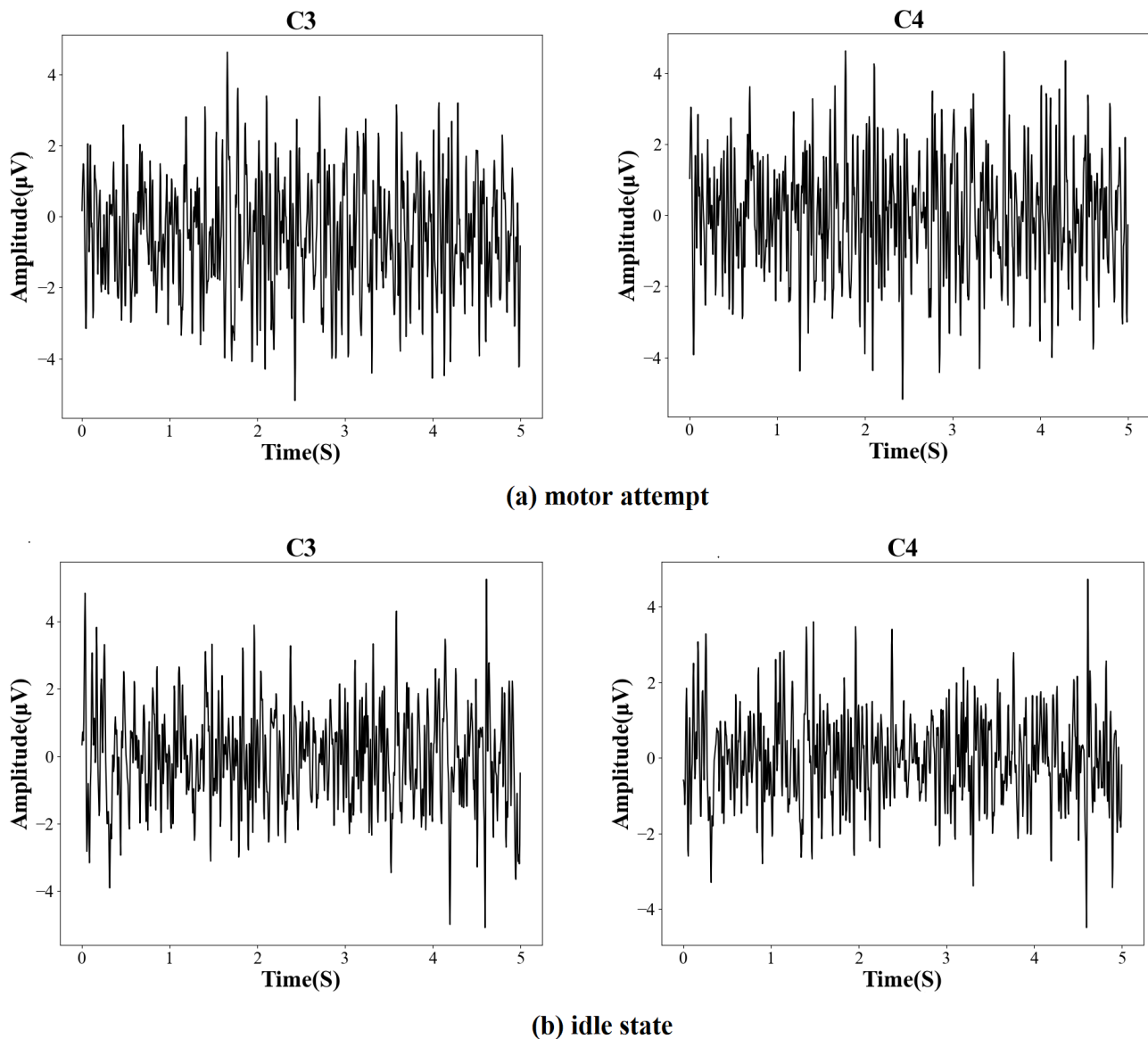


Figure 2. Examples of preprocessed EEG signals from different brain activities.

2.3. Overlapping Time Window

The performance of DL models is very heavily dependent on the quantity of data involved in the training. Due to the difficulty of collecting MA data, the amount of data collected for individual patients is small. In existing work, a promising approach is to split the individual signals into multiple subsignals for training the model [19,20,22]. We propose a data augmentation method based on an overlapping time window for increasing the number of instances during training. The raw EEG data were segmented by overlapping

windows; each data window served as an independent instance. The number of windows was controlled by two parameters: the time window length L and the overlap rate O . For the original input of experimental data $X^i = [x_1, x_2, \dots, x_T] \in R^{C \times T}$, i represented the type of task, C represented the channel, and T represented the sampling point; in this experiment, $C = 31$ and $T = 1000$. Given the parameters O, L . The raw data were segmented into $D_{L,O}^i$.

$$D_{L,O}^i = \{X_1^i, X_2^i, \dots, X_n^i\} \in R^{n \times C \times L} \quad (1)$$

$$X_s^i = \{x_t, x_{t+1}, \dots, x_{t+L} \mid t = 1 + (s-1)LO\} \quad (2)$$

$n = \frac{T-L}{LO} + 1$, which denotes that the original data were sliced into n time segments. The segmented data and the original data had the same task label i . When $n = 1$, the data were not segmented. We divided the original dataset into a training set and a test set. For the training set, each signal X_{train} with a length of 1000 was divided into 32 windows ($L = 60, O = 0.5$). In this way, the signals in each window were used as instances to train the model. In the testing phase, for each signal X_{test} with a length of 1000 in the test dataset, we divided it into 32 windows using the same approach. The data from these windows were fed into the trained model, and the classification results were obtained on these windows. After that, multiple window classification results of windows were fused into one decision for X_{test} by using the vote-counting strategy (VS). In addition, we also designed another method for classifying X_{test} . We combined the features of different windows from the last hidden layer of the model by summing, and the combined feature was fed into the softmax layer for classification, which is called the feature fusion strategy (FFS) in this paper. “&VS” and “&FFS” refer to strategies for validating trained models on the test set using voting and feature fusion, respectively.

2.4. Graph Isomorphism Network Model

2.4.1. Graph Data Construction

The aim of graph neural networks (GNNs) is to use graph structure data and node features as input to learn a representation of the node (or graph) for relevant tasks [23]. Because EEG data are easily converted to graph structure data, several studies have investigated GNNs applied to EEG signal-based tasks [17,24–27]. An important aspect of using a GNN to classify EEG signals is building graph data, the original data first need to be converted into graph structure data. The EEG signal of a window can be defined as $G = (V, E)$, where V and E represent the sets of nodes and edges, respectively. In this experiment, we treated the individual channel as a node and the closed channels as connected edges. Specifically, the average Euclidean distance d between the CZ channel and the other channels was calculated, and the two channels with an electrode distance less than d were treated as connected.

2.4.2. Graph Isomorphism Network

Most GNNs complete the graph classification process through a strategy of aggregating information from neighbors. Formally, node updates and the graph embedding h_G are obtained using the following formula.

$$a_v^{(k)} = \text{AGGREGATE}^{(k)}\left(\left\{h_u^{(k-1)} \mid u \in N(v)\right\}\right) \quad (3)$$

$$h_v^{(k)} = \text{COMBINE}^{(k)}\left(h_v^{(k-1)}, a_v^{(k)}\right) \quad (4)$$

$$h_G = \text{READOUT}\left(\left\{h_v^{(K)} \mid v \in G\right\}\right) \quad (5)$$

where $h_v^{(k)}$ is the feature vector of node v at the k -th iteration layer, $h_v^{(0)}$ represents node input, and $N(v)$ is a set of nodes adjacent to v . AGGREGATE and COMBINE represent the aggregation of information about neighbors and the aggregation of information about

oneself and neighbors, respectively. The choices of $\text{AGGREGATE}^{(k)}(\cdot)$ and $\text{COMBINE}^{(k)}(\cdot)$ in GNNs are crucial. The output h_G aggregates node features via the READOUT function at the final iteration.

In this study, we used a graph isomorphism network (GIN) model for classifying EEG signals. The GIN network is a kind of GNN and uses the summation method to complete AGGREGATE, COMBINE, and READOUT [28]. Following the literature [28], we used the following formula for feature updates of node features:

$$h_v^{(k)} = \text{MLP}^{(k)} \left(\left(1 + \epsilon^{(k)} \right) \cdot h_v^{(k-1)} + \sum_{u \in N(v)} h_u^{(k-1)} \right) \quad (6)$$

MLP represents multilayer perceptrons, and ϵ is a parameter that can be trained. We tuned the hyperparameters through a grid search over the training set in this experiment. The search space ranges for the network depth and the number of neurons in GIN were defined as {1, 2, 3, 4} and {32, 64, 128, 256}, respectively. After hyperparameter optimization, the network depth k and the number of neurons were set to 2 and 256, respectively. The learned graph embedding h_G passes through two fully connected layers to output the final feature representation. This experiment focused on the binary classification problem, so the number of neurons in the final fully connected layer was 2. In this study, the output layers of the three models were the softmax layer, and the loss functions were all set to the cross-entropy loss function.

2.5. CNN

Convolutional neural networks (CNNs) are considered to be one of the most successful deep learning models and have been widely used for feature extraction of EEG signals [24,29,30]. The CNN is a deep feed-forward neural network that includes crucial convolutional operations. Compared to traditional neural networks, CNNs reduce the training parameters by local sensing and weight sharing. Each convolutional layer consists of multiple convolutional kernels of the same size for feature extraction. The mathematical description of the convolutional operation is as follows:

$$y_{mn} = f \left(\sum_{j=0}^{I-1} \sum_{i=0}^{I-1} x_{m+i, n+j} w_{ij} + b \right) \quad (7)$$

where x represents the matrix on which the convolution operation is performed, and y is the output of the convolution. I, J corresponds to the size of the convolution kernel w , b represents a bias, and f is the activation function, which was *ReLU* in this study. The grid search ranges of the parameters were defined as follows: number of convolution layers and max-pooling layers {1, 2, 3, 4}, length of the convolutional kernels {2, 3, 4, 5}, number of convolutional kernels {16, 32, 64, 128}, and number of neurons in the fully connected layer {32, 64, 128, 256}. The optimized model structure in this study consisted of 3 convolutional layers and 3 max pooling layers. The size of the convolutional kernels was $4 * 4$, $2 * 2$, and $2 * 2$, and the number of convolutional kernels was 32, 64, and 128, respectively. After completing the pooling of the final layer, we flattened the extracted features and fed them into a fully connected layer fc_1 , which contained 128 neurons. Finally, the output of the fully connected layer fc_1 passed through the *ReLU* activation function and another fully connected layer fc_2 to output the final 2-dimensional representation feature.

2.6. LSTM

Due to the long duration of the patient performing the task, some useful features still needed to be retained despite the long interval. LSTM can retain the motor intention of EEG signals that are both long and short. LSTM networks are a modified version of recurrent neural networks (RNNs) [31]. Based on RNNs, the LSTM added a multiple gate structure (forget gate f_t , input gate i_t , and output gate o_t) for updating the cell state. The LSTM

network layer contains the cell state C_t , which represents the cell information stored at time t . The data features x_t at time t , the hidden features h_{t-1} at moment $t - 1$, and the cell state C_{t-1} , were fed into the LSTM nodes, which were processed by gates to output the hidden state and cell state at the next moment. The calculations are as follows.

$$f_t = \text{sigmoid}(w_f[x_t, h_{t-1}] + b_f) \quad (8)$$

$$i_t = \text{sigmoid}(w_i[h_{t-1}, x_t] + b_i) \quad (9)$$

$$\bar{C}_t = \tanh(w_c[h_{t-1}, x_t] + b_c) \quad (10)$$

$$C_t = f_t * C_{t-1} + i_t * \bar{C}_t \quad (11)$$

where w_f and b_f indicate the weight and bias of the forget gate, respectively. The sigmoid function in the forget gate determines which messages need to be deleted. The corresponding input gate i_t determines which information to retain, and w_i and b_i indicate the weight and bias of the input gate. $\bar{C}_t$ represents the candidate hidden state, and w_c and b_c correspond to the weight and bias, respectively. The output of the forget gate and the input gate are jointly calculated to obtain the cell state value C_t at the current moment.

Finally, the current cell state C_t and the output o_t of the output gate are calculated as follows to obtain the current hidden state h_t .

$$o_t = \text{sigmoid}(w_o[h_{t-1}, x_t] + b_o) \quad (12)$$

$$h_t = o_t * \tanh(C_t) \quad (13)$$

where w_o and b_o are the weight and bias of the output gate, respectively. For a multilayer LSTM model, the hidden state h_t at moment t of the previous layer is used as the input of the next network layer at moment t . The number of LSTM hidden layer units was determined by the time window length, and each time point was a unit. In this study, we employed a two-layer LSTM, and we fed the hidden state at the last moment of the last layer into a fully connected layer to output the final feature representation. The hidden state features of the LSTM perform a grid search in the range {32, 64, 128, 256}, with an optimized feature size of 128.

2.7. Evaluation Procedures

One of the most important aspects of BCI is accuracy. To test the effectiveness of different methods, we used 3-fold cross-validation. For one session, the data were randomly divided into a training set containing 60 trials and a test set containing 30 trials, with a ratio of 2:1. We first optimized hyperparameters on the training set via grid search, with 90% of the data used to train the model and 10% to validate the performance of the hyperparameters and choose the model structure with the highest average accuracy. The data ratio between the two task categories was always 1:1 in the different sets. After completing hyperparameter optimization, we trained the final classification models using all the training data. The average accuracy of each fold was used to evaluate the performance of the model. In addition, we used the information transfer rate (ITR) to evaluate the performance of the BCI [32]. The units of ITR are bits/min, which are calculated from Equation (14). N is the number of task types, which is set to 2 in this study, P is the accuracy rate, and T is the time during the task (60 s).

$$B = \left\{ \log_2 N + P \log_2 P + (1 - P) \log_2 \left[\frac{1 - P}{N - 1} \right] \right\} \times \frac{60}{T} \quad (14)$$

3. Results and Discussion

3.1. Overall Performance

In this study, the average classification results for seven subjects are reported in Table 2. The results listed include six methods and the accuracies achieved with the same segmen-

tation strategy ($L = 60$, $O = 0.5$). The results showed that the LSTM&FFS achieved the highest mean accuracy of 90.1% for all subjects, while the GIN&VS had the lowest accuracy of 81.9%. One of the explanations for differences in accuracy between different methods could be different types of extracted features. To verify that our method is superior to the existing method, we compared our methods with the traditional algorithms [14,33]. After comparison, compared with CSP and FBCSP, the six methods in this study yielded an average improvement of 11.71% and 23.01% in regard to average accuracy. The results indicated that the methods can learn distinctive features of multiple windows for classification and that these features improved classifier performance. To analyze the impact of age on the algorithm, patients were divided into two age groups based on the median age (40 years). Compared to the group aged <40 , the group aged ≥ 40 showed higher accuracy on different algorithms, and the results suggest that the patient's age may be a factor in the accuracy of the classification.

To identify the performance of different models, we recorded the training loss and validation accuracy during model training for Subject 1 in Figure 3. The loss in the CNN and LSTM training set dropped to 0.2 after 100 iterations and the top accuracy of the test sets converged to approximately 93% and 98%, respectively. A faster convergence was observed in GIN, but the accuracy was relatively low. Figure 4 illustrates the training time and the number of training parameters for different models. Although the training parameters of the LSTM model were not the highest, training of the model required more time. This may partially be because of the structure of the LSTM model [34], which could not complete parallel computing in the training process. For real-world applications, the choice of method can consider multiple factors of accuracy and computational complexity.

Paired-sample t-tests were used to determine whether the difference among individual methods was statistically significant. The results are presented in Table 3. Significant differences were found across multiple methods at the 0.05 significance level, and the difference between LSTM&VS and LSTM&FFS was not significant at the 0.01 significance level. Although the performance was different in the six methods, the overall accuracy was high.

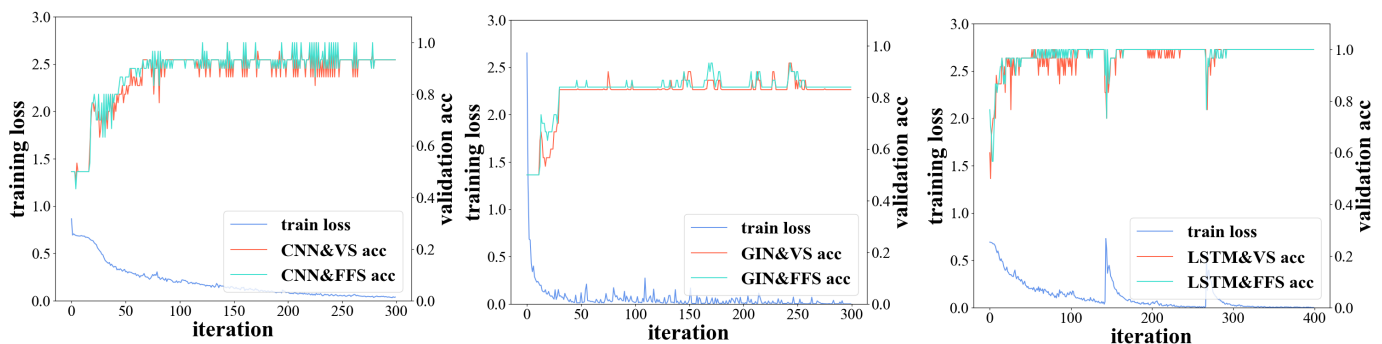


Figure 3. Losses of the training sets for deep learning models and classification accuracies of test sets for different methods.

Table 2. The overall comparison results of average classification performance.

Subjects	Accuracy							
	CSP	FBCSP	GIN&VS	LSTM&VS	CNN&VS	GIN&FFS	LSTM&FFS	CNN&FFS
sub1	0.798 ± 0.079	0.614 ± 0.096	0.914 ± 0.061	0.979 ± 0.025	0.944 ± 0.059	0.931 ± 0.049	0.980 ± 0.017	0.934 ± 0.058
sub2	0.700 ± 0.064	0.578 ± 0.096	0.722 ± 0.069	0.772 ± 0.106	0.741 ± 0.065	0.746 ± 0.060	0.806 ± 0.111	0.757 ± 0.057
sub3	0.771 ± 0.077	0.627 ± 0.094	0.902 ± 0.075	0.938 ± 0.044	0.932 ± 0.047	0.919 ± 0.073	0.949 ± 0.042	0.940 ± 0.046
sub4	0.804 ± 0.072	0.635 ± 0.157	0.938 ± 0.038	0.993 ± 0.011	0.966 ± 0.024	0.952 ± 0.030	0.994 ± 0.010	0.970 ± 0.016
sub5	0.724 ± 0.054	0.644 ± 0.057	0.711 ± 0.066	0.808 ± 0.071	0.762 ± 0.076	0.757 ± 0.073	0.841 ± 0.068	0.769 ± 0.085
sub6	0.687 ± 0.040	0.630 ± 0.053	0.744 ± 0.055	0.801 ± 0.037	0.764 ± 0.058	0.777 ± 0.055	0.825 ± 0.029	0.755 ± 0.050
sub7	0.707 ± 0.082	0.677 ± 0.090	0.798 ± 0.078	0.899 ± 0.066	0.845 ± 0.085	0.842 ± 0.081	0.909 ± 0.062	0.856 ± 0.084
Mean	0.742	0.629	0.819	0.884	0.851	0.846	0.901	0.854
Group1	0.736 ± 0.056	0.629 ± 0.015	0.790 ± 0.108	0.865 ± 0.099	0.823 ± 0.105	0.822 ± 0.095	0.884 ± 0.088	0.820 ± 0.100
Group2	0.746 ± 0.050	0.630 ± 0.041	0.840 ± 0.099	0.901 ± 0.094	0.871 ± 0.101	0.865 ± 0.092	0.915 ± 0.080	0.881 ± 0.096

Group1, age < 40; Group2, age ≥ 40.

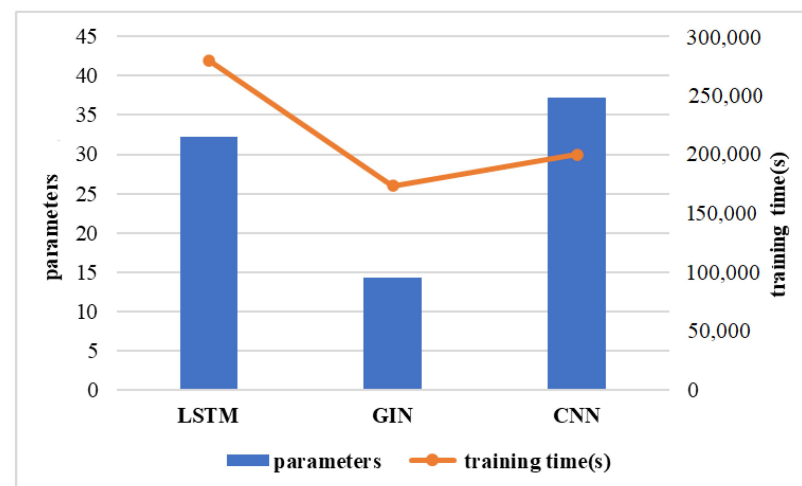


Figure 4. The number of parameters and running times for different models.

Table 3. Paired-sample *t*-test results of different methods.

Method	Comparsion Method	T	df	Sig. (2-Tailed)
GIN&VS	GIN&FFS	−5.451	6	0.002
	CNN&FFS	−5.345	6	0.002
	LSTM&FFS	−7.528	6	<0.001
	CNN&VS	−6.705	6	0.001
	LSTM&VS	−7.140	6	<0.001
CNN&VS	GIN&FFS	1.146	6	0.295
	CNN&FFS	−0.999	6	0.356
	LSTM&FFS	−5.976	6	0.001
	LSTM&VS	−5.805	6	0.001
LSTM&VS	GIN&FFS	6.823	6	<0.001
	CNN&FFS	4.333	6	0.005
	LSTM&FFS	−3.527	6	0.012
GIN&FFS	CNN&FFS	−1.501	6	0.184
	LSTM&FFS	−8.477	6	<0.001
CNN&FFS	LSTM&FFS	−5.400	6	0.002

3.2. Effects of Window Size

To investigate the effect of window size, we experimented with multiple window sizes and evaluated the performances of the models in the corresponding window. The sizes of the comparison time windows were 1000, 200, 100, 90, 80, 70, and 60. Figure 5 presents the average accuracy of the six methods for the different windows. The lowest accuracy was achieved when the window size was 1000, which means without data augmentation. The most likely reason for this phenomenon is probably because the models overfit fewer training data. We illustrate the performance of each method for three window sizes in Table 4. The LSTM&VS accuracy was 90.3% when the window size was 70. However, the LSTM model showed only a classification accuracy of 65.4% without using a time window. The difference in accuracies occurred because the different window sizes changed the lengths of the input time series. In addition, according to the results, when the window size was 70, the average accuracy of each method was higher than that of the method with a window size of 100. This may indicate that relatively small sizes of windows had better performance.

Table 4. Comparison of the accuracies of different methods with time windows of 60, 70, and 1000.

Window Length	Method	Accuracy							
		Sub1	Sub2	Sub3	Sub4	Sub5	Sub6	Sub7	Mean
70	GIN&VS	0.929 ± 0.061	0.764 ± 0.069	0.915 ± 0.075	0.942 ± 0.038	0.760 ± 0.066	0.781 ± 0.055	0.858 ± 0.078	0.850
	GIN&FFS	0.924 ± 0.049	0.755 ± 0.060	0.912 ± 0.073	0.947 ± 0.030	0.755 ± 0.073	0.782 ± 0.054	0.843 ± 0.081	0.845
	CNN&VS	0.94 ± 0.039	0.766 ± 0.076	0.931 ± 0.047	0.967 ± 0.021	0.777 ± 0.059	0.789 ± 0.054	0.871 ± 0.083	0.864
	CNN&FFS	0.939 ± 0.040	0.754 ± 0.079	0.929 ± 0.049	0.960 ± 0.018	0.761 ± 0.053	0.760 ± 0.059	0.856 ± 0.095	0.851
	LSTM&VS	0.985 ± 0.018	0.808 ± 0.101	0.951 ± 0.041	0.997 ± 0.008	0.844 ± 0.072	0.819 ± 0.036	0.917 ± 0.066	0.903
	LSTM&FFS	0.984 ± 0.014	0.801 ± 0.101	0.948 ± 0.038	0.995 ± 0.008	0.842 ± 0.078	0.812 ± 0.047	0.917 ± 0.078	0.900
100	GIN&VS	0.925 ± 0.046	0.764 ± 0.058	0.916 ± 0.065	0.929 ± 0.039	0.746 ± 0.056	0.793 ± 0.053	0.844 ± 0.074	0.845
	GIN&FFS	0.919 ± 0.053	0.755 ± 0.048	0.919 ± 0.069	0.935 ± 0.039	0.745 ± 0.070	0.784 ± 0.056	0.846 ± 0.080	0.843
	CNN&VS	0.938 ± 0.058	0.759 ± 0.057	0.923 ± 0.047	0.941 ± 0.030	0.755 ± 0.052	0.762 ± 0.053	0.852 ± 0.075	0.847
	CNN&FFS	0.925 ± 0.056	0.744 ± 0.064	0.920 ± 0.055	0.946 ± 0.029	0.748 ± 0.053	0.734 ± 0.057	0.830 ± 0.096	0.835
	LSTM&VS	0.982 ± 0.020	0.799 ± 0.095	0.943 ± 0.039	0.995 ± 0.008	0.807 ± 0.077	0.893 ± 0.038	0.816 ± 0.078	0.891
	LSTM&FFS	0.986 ± 0.017	0.792 ± 0.098	0.942 ± 0.040	0.995 ± 0.010	0.814 ± 0.084	0.898 ± 0.046	0.815 ± 0.089	0.892
1000	GIN	0.810 ± 0.057	0.732 ± 0.062	0.814 ± 0.091	0.797 ± 0.051	0.696 ± 0.067	0.700 ± 0.037	0.731 ± 0.057	0.754
	CNN	0.689 ± 0.084	0.625 ± 0.058	0.763 ± 0.013	0.643 ± 0.066	0.634 ± 0.051	0.612 ± 0.062	0.638 ± 0.067	0.658
	LSTM	0.674 ± 0.061	0.630 ± 0.051	0.719 ± 0.111	0.633 ± 0.075	0.644 ± 0.042	0.629 ± 0.042	0.646 ± 0.064	0.654

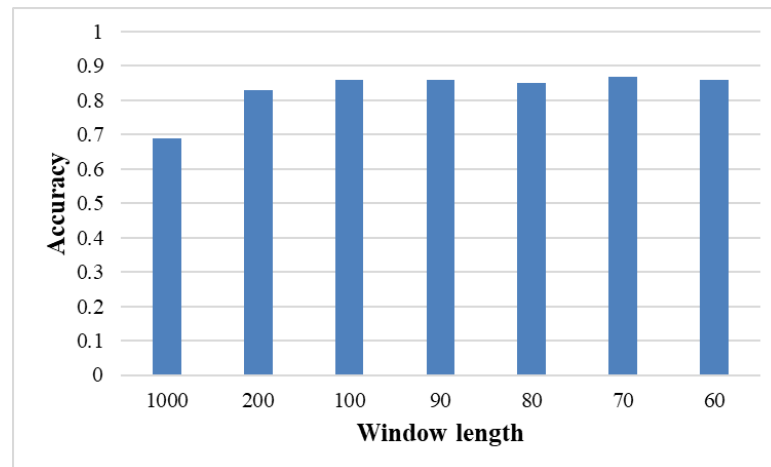


Figure 5. Accuracies for all methods with different window lengths, each column represents the average of the six method accuracies.

3.3. Generic Performance of BCI

It is difficult to compare different BCI systems since there are many aspects that can influence the performance of BCI, such as input, preprocessing, and outputs. The ITR is a widely and generally accepted standard by which the performance of different BCI systems can be compared [35]. Figure 6 illustrates the distribution of ITRs for the sessions. The average ITR for all seven subjects was 10.72 ± 4.82 bits/min. Several subjects (1, 3, 4 and 7) reached the highest ITR with 12 bits/min. Subject 7 had the lowest ITR of 1.73 bits/min. For the ITR of motor attempts, fewer results have been reported. Khalaf et al. obtained an average ITR of 40.83 bits/min for a four-class task [36]. Zeng et al. achieved the highest ITR of 24 bits/min during ankle rehabilitation robot training [37]. In this study, the value of ITR is negatively correlated with T , and the value of ITR is limited by task time.

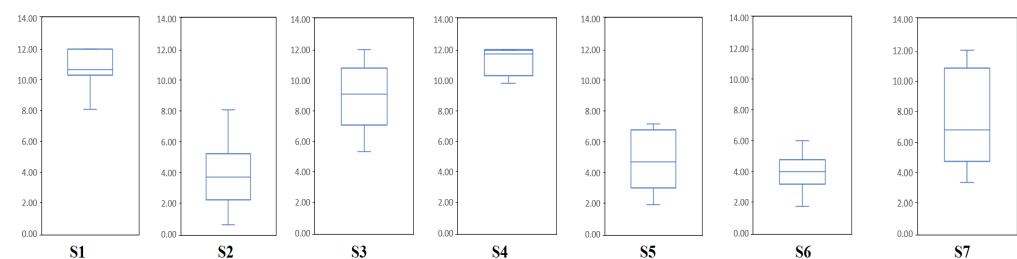


Figure 6. Boxplots represent the ITRs of different sessions for each subject. Each box plot includes 12 sessions of data. The upper and lower lines represent the maximum and minimum ITRs, respectively. The lines in the boxplot represent the median ITR.

3.4. The Visualization of Feature Distribution

To investigate the validity of the time window. A visualization technique called TSNE [38] was used to downscale the learned features for visualization. Figure 7 shows the distribution of features for different time windows. The different colored scatter points in the figure indicate the different task types, and each scatter represents the extracted feature by one window. We observed that the features extracted by methods with time windows were easier to classify. After training, multiple EEG signals of time segments were well identified. However, some EEG signals from different windows were hard to identify. In addition, it was also found that among CNN, GIN, and LSTM, LSTM performed best in feature extraction, which had fewer segments that could not be distinguished. To further observe the LSTM performance, we constructed the LSTM&VS confusion matrices of the seven subjects in Figure 8. The correct classification accuracies are shown on the diagonal cells. It can be seen that the LSTM&VS accuracy in each task

was similar for individual subjects. The results demonstrated that LSTM&VS obtained good overall performance.

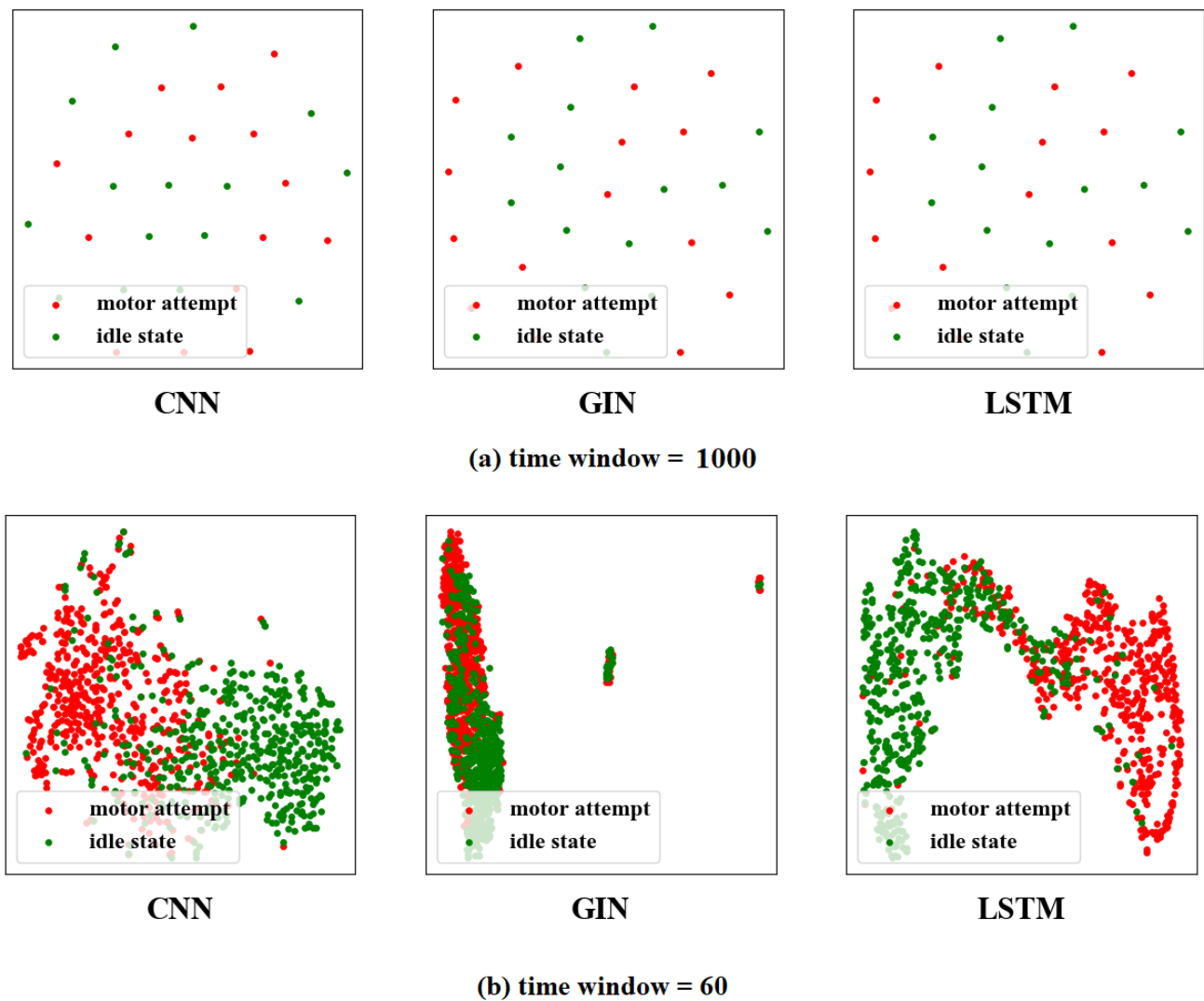


Figure 7. Feature visualization of different models for Subject 1. For time window (a), each scatter represents a feature extracted on one trail, and for time window (b), each scatter represents a feature extracted on one time window of one trail.

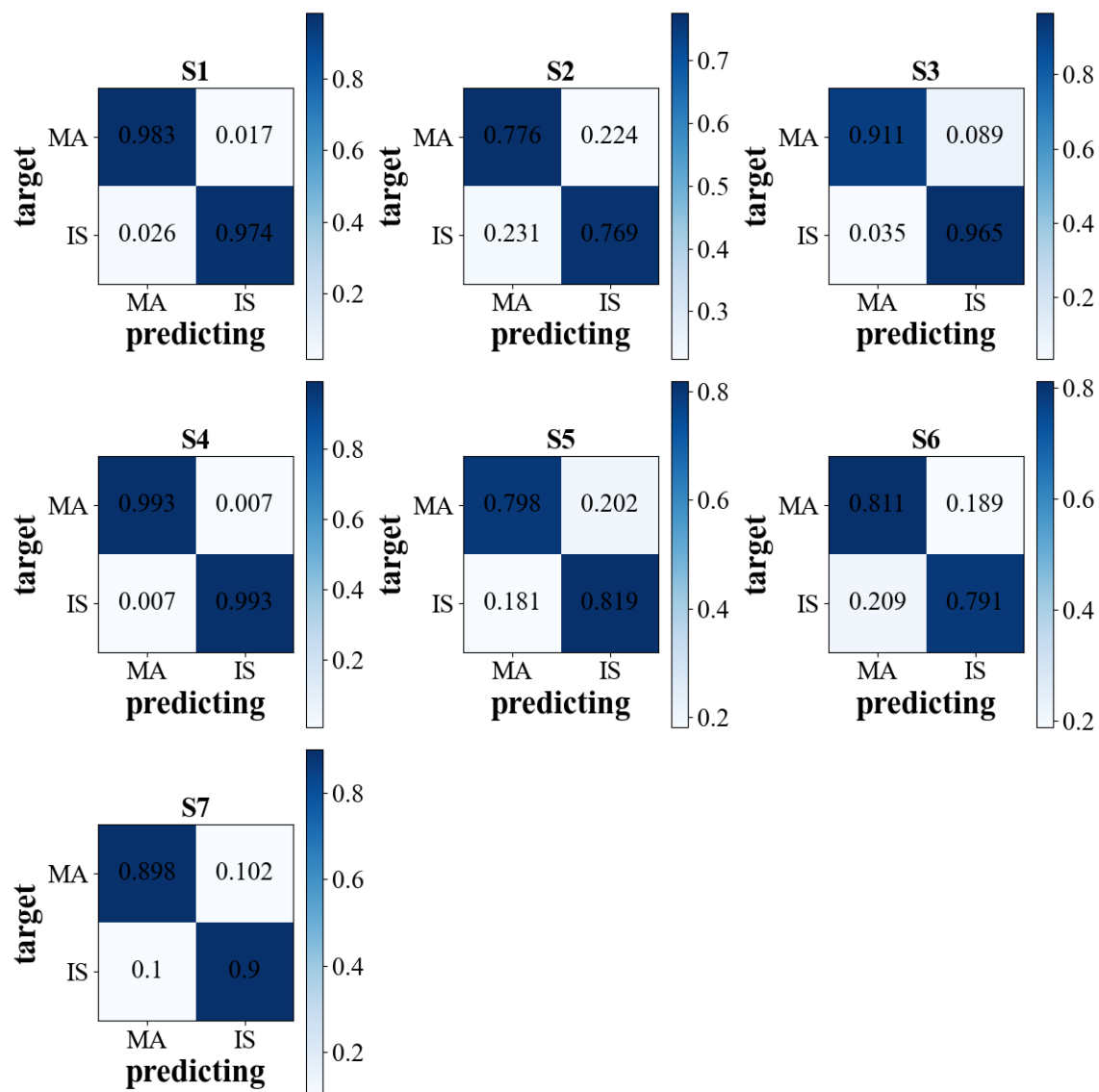


Figure 8. The confusion matrices for all subjects with the LSTM&VS method.

3.5. The Impact of the Number of Network Layers

The number of network layers usually affected the model performance. Table 5 shows the accuracy of models with different numbers of network layers. Compared with other settings, the accuracy of 1 layer was lower. In addition, the accuracies of the CNN&VS and CNN&FFS were more likely to be influenced by the number of network layers, while the accuracies of LSTM&VS and LSTM&FFS were more stable.

Table 5. Accuracy comparison between different numbers of network layers.

Conv Layers	Accuracy					
	GIN&VS	CNN&VS	LSTM&VS	GIN&FFS	CNN&FFS	LSTM&FFS
1 layer	0.803 ± 0.082	0.821 ± 0.091	0.882 ± 0.079	0.834 ± 0.074	0.791 ± 0.072	0.900 ± 0.065
2 layers	0.819 ± 0.085	0.844 ± 0.086	0.884 ± 0.078	0.846 ± 0.075	0.846 ± 0.068	0.901 ± 0.066
3 layers	0.816 ± 0.078	0.851 ± 0.083	0.883 ± 0.068	0.843 ± 0.082	0.854 ± 0.080	0.896 ± 0.069
4 layers	0.806 ± 0.089	0.844 ± 0.088	0.882 ± 0.079	0.821 ± 0.078	0.847 ± 0.069	0.900 ± 0.072
Mean	0.810	0.840	0.882	0.836	0.835	0.900

The highest classification accuracy for a given method are bold marked.

3.6. The Visualization of Accuracy on Time Window

In this section, to analyze the classification differences in each time window, we conducted statistics and visualized the classification results of the EEG signals on the windows by using the LSTM&VS method. Figure 9 shows the sequence of classification results. Each row represents a session, and each column represents the classification statistics of the time window, in which the time windows with higher accuracy are highlighted. As seen in the figure, the distribution of accuracy across the time windows differs in patients. For Subject 3, the time window with higher classification accuracy appeared in the first window, and the window with lower classification accuracy appeared in the last window. The difference in sequence probably occurred because the appearances of the discriminative motor intentions were random during the MA experiment.

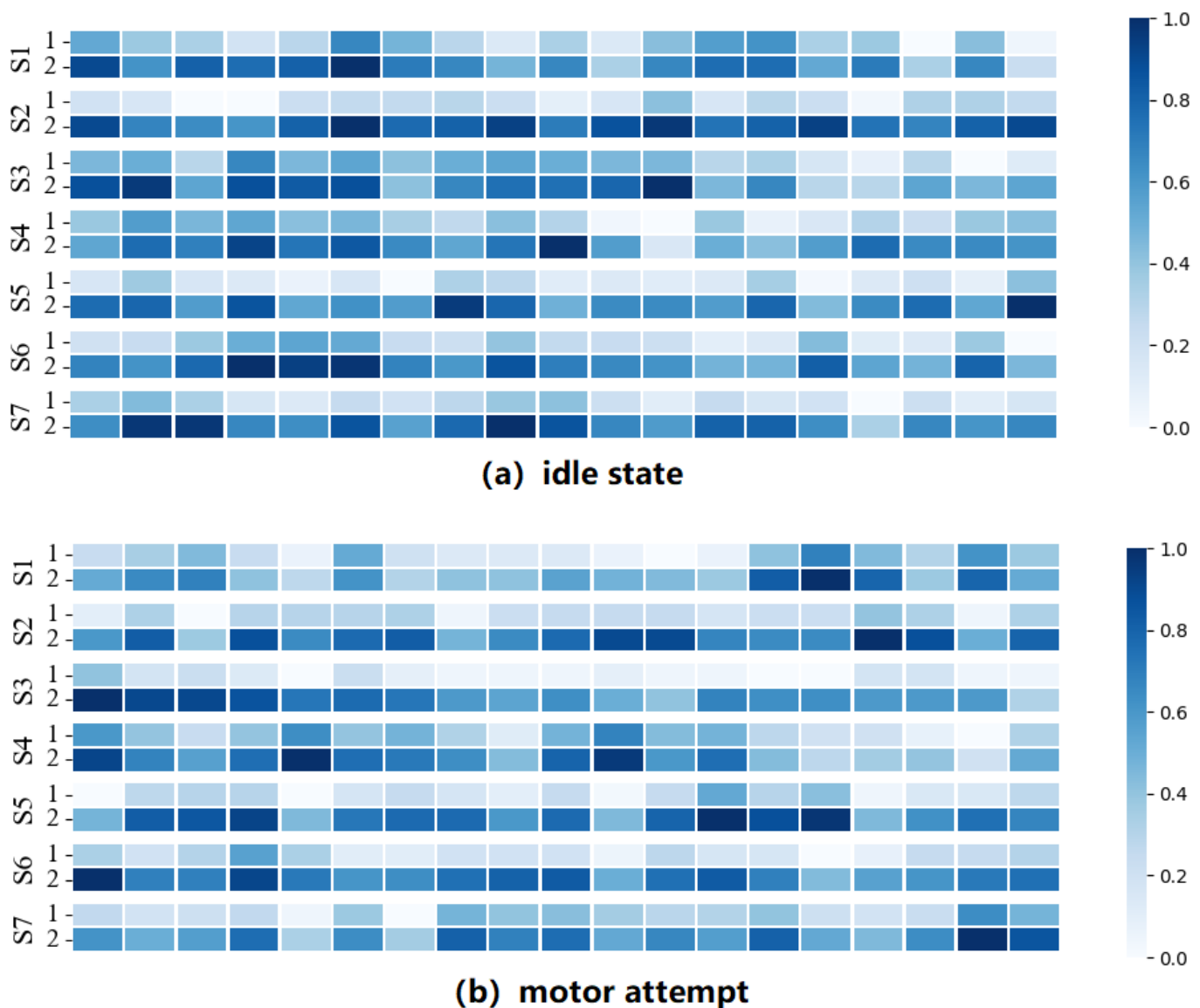


Figure 9. Visualization of classification results learned by LSTM&VS. In each state, two different sessions (1, 2) of seven subjects are visualized. Each row represents a session and each square represents the correct result for classification in a time window.

3.7. Study of Cortical Activity on the Time Window

To further investigate the differences in the EEG signals of time segments with different classification accuracies, we used power spectral density topography to represent

the frequency domain information of brain signals. Figure 10 illustrates the topography of alpha power for two tasks. Based on the classification accuracy of the time segments in Figure 9, four different time windows were selected for visualization. For the motor attempt task, the PSD of patients was higher in the frontal lobe. Several studies have indicated that stroke can affect the brain function in the frontal lobe [39,40]. The observations may suggest that motor attempts of patients were associated with the frontal cortical regions, which is consistent with a previous study [41]. Channel information from frontal regions may be important for identifying the brain's motor intentions. When the EEG signals in the frontal lobe are not significant enough, it may contribute to lower classification accuracy.

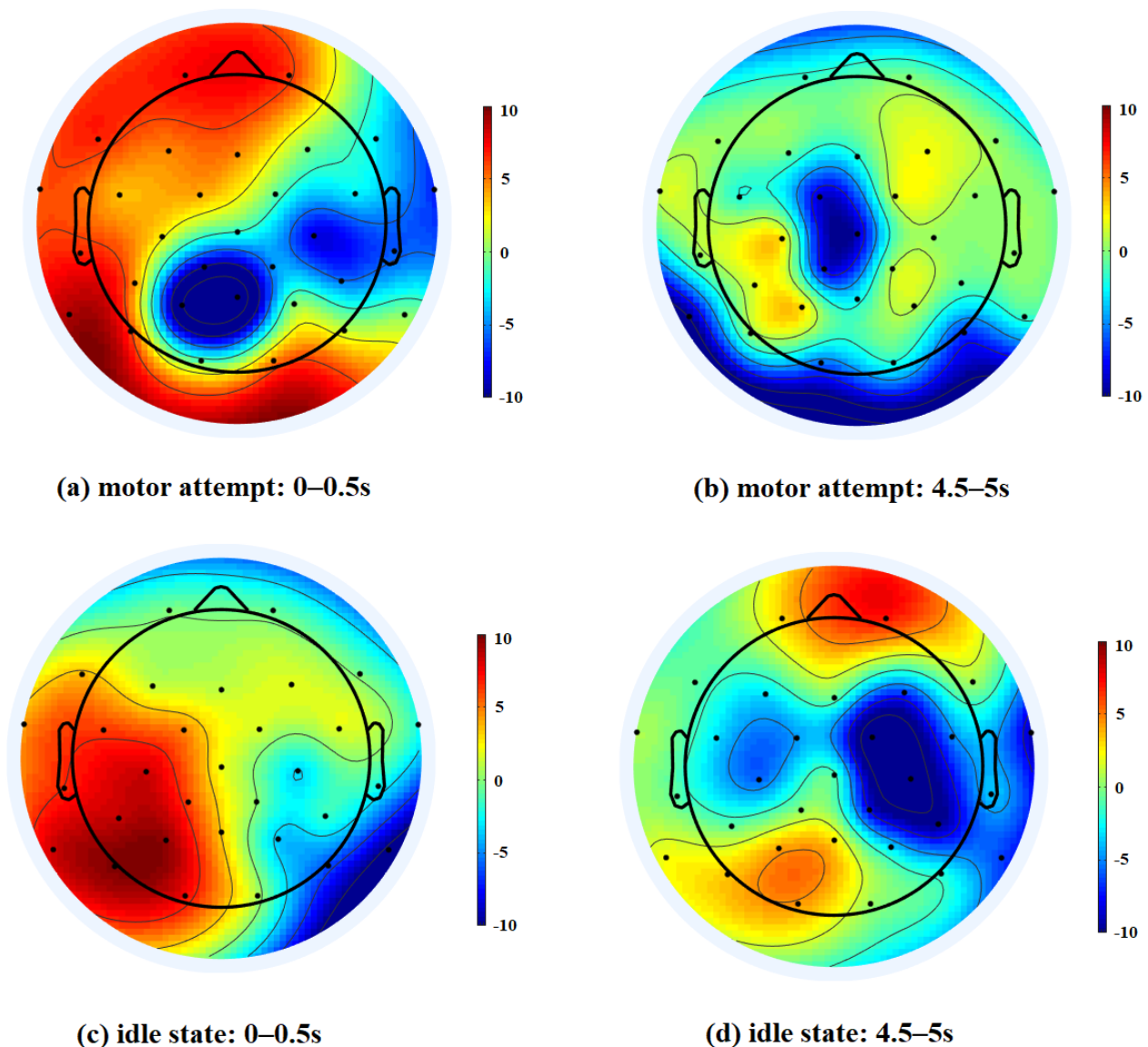


Figure 10. The visualization of the averaged topography of the PSD over the alpha band in different time windows. Session 3 with two task types for Subject 3 is visualized.

3.8. Limitations in Current Work

The findings in this study are limited by the quantity of data collected, and it is difficult to determine the quantity of data that can be classified well without using a time window. In addition, the optimal filter band and model hyperparameters were not selected according to the subjects, which may limit the ability of the models in different patients. In future

work, we will investigate the use of feature engineering for reducing the dimensionality of the model inputs. In addition, the fixed starting point for time window sampling may reduce the performance of the BCI system [42]. Therefore, we will optimize the set of time windows by using a window selection algorithm in the feature work.

4. Conclusions

This study showed that for classification tasks during BCI-based stroke rehabilitation, deep learning algorithms based on overlapping time windows achieved good accuracy. It may support improvements in the performance of brain–computer interfaces to generate accurate neurofeedback. One of the more significant findings to emerge from this study is that the distribution of classification results differed across the time windows of the subjects, and it means that there is a possibility of improving classification performance by choosing different windows for classification for different subjects. Therefore, future work can expand on the selection of the time window.

Author Contributions: Conceptualization, L.C.; methodology, L.C., H.W., S.C., C.F. and Y.D.; software, S.C. and J.J.; data curation, S.C.; writing—original draft preparation, L.C. and H.W.; writing—review and editing, L.C., H.W., C.Z. and Y.D.; visualization, L.C. and H.W.; supervision, L.C., C.Z. and Y.D.; project administration, L.C.; L.C., H.W. and S.C. contributed equally. All authors have read and agreed to the published version of the manuscript.

Funding: The work was supported by the National Key R&D Program of China under Grants 2018YFC2002300, 2018YFC2002301, the National Natural Science Young Foundation of China (Grant No. 62102242 & 62103258), Shanghai Education Research Program (Grant No. C2022152), Shanghai Science and Technology Innovation Action Plan (22YF1404200) and Project of Wuxi Health Commission (Grant No. Z2022012, MS201944 and T201906).

Institutional Review Board Statement: The study was conducted in accordance with the Declaration of Helsinki, and approved by the Institutional Review Board of Ethics Committee of Huashan Hospital (Approval no.: 18518-201111KY2017-005).

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Broussalis, E.; Killer, M.; McCoy, M.; Harrer, A.; Trinka, E.; Kraus, J. Current therapies in ischemic stroke. Part A. Recent developments in acute stroke treatment and in stroke prevention. *Drug Discov. Today* **2012**, *17*, 296–309. [\[CrossRef\]](#) [\[PubMed\]](#)
2. Thieme, H.; Mehrholz, J.; Pohl, M.; Behrens, J.; Dohle, C. Mirror therapy for improving motor function after stroke. *Stroke* **2013**, *44*, e1–e2. [\[CrossRef\]](#) [\[PubMed\]](#)
3. Saposnik, G.; McIlroy, W.E.; Teasell, R.; Thorpe, K.E.; Bayley, M.; Cheung, D.; Mamdani, M.; Hall, J.; Cohen, L.G. Effectiveness of virtual reality using Wii gaming technology in stroke rehabilitation: A pilot randomized clinical trial and proof of principle. *Stroke* **2010**, *41*, 1477. [\[CrossRef\]](#) [\[PubMed\]](#)
4. Rimmer, J.H.; Wang, E. Aerobic Exercise Training in Stroke Survivors. *Top. Stroke Rehabil.* **2005**, *12*, 17–30. [\[CrossRef\]](#)
5. Ang, K.K.; Guan, C.; Phua, K.S.; Wang, C.; Zhou, L.; Tang, K.Y.; Ephraim Joseph, G.J.; Kuah, C.W.K.; Chua, K.S.G. Brain-computer interface-based robotic end effector system for wrist and hand rehabilitation: Results of a three-armed randomized controlled trial for chronic stroke. *Front. Neuroeng.* **2014**, *7*, 30. [\[CrossRef\]](#)
6. Cervera, M.A.; Soekadar, S.R.; Ushiba, J.; Millán, J.D.R.; Liu, M.; Birbaumer, N.; Garipelli, G. Brain-computer interfaces for post-stroke motor rehabilitation: A meta-analysis. *Ann. Clin. Transl. Neurol.* **2018**, *5*, 651–663. [\[CrossRef\]](#)
7. Mane, R.; Chouhan, T.; Guan, C. BCI for stroke rehabilitation: Motor and beyond. *J. Neural. Eng.* **2020**, *17*, 041001. [\[CrossRef\]](#)
8. Pfurtscheller, G.; Neuper, C. Motor imagery activates primary sensorimotor area in humans. *Neurosci. Lett.* **1997**, *239*, 65–68. [\[CrossRef\]](#)
9. Philip, G.R.; Daly, J.J.; Príncipe, J.C. Topographical measures of functional connectivity as biomarkers for post-stroke motor recovery. *J. Neuroeng. Rehabil.* **2017**, *14*, 67. [\[CrossRef\]](#)
10. Xiao, X.; Xu, M.; Jin, J.; Wang, Y.; Jung, T.P.; Ming, D. Discriminative canonical pattern matching for single-trial classification of ERP components. *IEEE Trans. Biomed. Eng.* **2019**, *67*, 2266–2275. [\[CrossRef\]](#)

11. Ramoser, H.; Muller-Gerking, J.; Pfurtscheller, G. Optimal spatial filtering of single trial EEG during imagined hand movement. *IEEE Trans. Rehabil. Eng.* **2000**, *8*, 441–446. [[CrossRef](#)] [[PubMed](#)]
12. Ang, K.K.; Chin, Z.Y.; Zhang, H.; Guan, C. Filter bank common spatial pattern (FBCSP) in brain-computer interface. In Proceedings of the 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), Hong Kong, China, 1–8 June 2008; pp. 2390–2397.
13. Geethanjali, P.; Mohan, Y.K.; Sen, J. Time domain feature extraction and classification of EEG data for brain computer interface. In Proceedings of the 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery, Chongqing, China, 29–31 May 2012; pp. 1136–1139.
14. Chen, S.; Shu, X.; Wang, H.; Ding, L.; Fu, J.; Jia, J. The differences between motor attempt and motor imagery in brain-computer interface accuracy and event-related desynchronization of patients with hemiplegia. *Front. Neurobot.* **2021**, *15*, 706630. [[CrossRef](#)] [[PubMed](#)]
15. Lin, P.J.; Jia, T.; Li, C.; Li, T.; Qian, C.; Li, Z.; Pan, Y.; Ji, L. CNN-Based Prognosis of BCI Rehabilitation Using EEG From First Session BCI Training. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2021**, *29*, 1936–1943. [[CrossRef](#)]
16. Liang, F.Y.; Zhong, C.H.; Zhao, X.; Castro, D.L.; Chen, B.; Gao, F.; Liao, W.H. Online adaptive and lstm-based trajectory generation of lower limb exoskeletons for stroke rehabilitation. In Proceedings of the 2018 IEEE International Conference on Robotics and Biomimetics (ROBIO), Kuala Lumpur, Malaysia, 12–15 December 2018; pp. 27–32.
17. Jin, J.; Sun, H.; Daly, I.; Li, S.; Liu, C.; Wang, X.; Cichocki, A. A Novel Classification Framework Using the Graph Representations of Electroencephalogram for Motor Imagery Based Brain-Computer Interface. *IEEE Trans. Neural. Syst. Rehabil. Eng.* **2021**, *30*, 20–29. [[CrossRef](#)]
18. Wang, F.; Zhong, S.h.; Peng, J.; Jiang, J.; Liu, Y. Data augmentation for eeg-based emotion recognition with deep convolutional neural networks. In *Proceedings of the International Conference on Multimedia Modeling*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 82–93.
19. A, I.U.; B, M.H.; B, E.U.H.Q.; B, H.A. An automated system for epilepsy detection using EEG brain signals based on deep learning approach. *Expert Syst. Appl.* **2018**, *107*, 61–71.
20. Zhang, G.; Davoodnia, V.; Sepas-Moghaddam, A.; Zhang, Y.; Etemad, A. Classification of hand movements from EEG using a deep attention-based LSTM network. *IEEE Sens. J.* **2019**, *20*, 3113–3122. [[CrossRef](#)]
21. Hartmann, K.G.; Schirrmester, R.T.; Ball, T. EEG-GAN: Generative adversarial networks for electroencephalographic (EEG) brain signals. *arXiv* **2018**, arXiv:1806.01875.
22. Truong, N.D.; Zhou, L.; Kavehei, O. Semi-supervised Seizure Prediction with Generative Adversarial Networks. In Proceedings of the 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Berlin, Germany, 23–27 July 2019.
23. Zhou, J.; Cui, G.; Hu, S.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; Sun, M. Graph neural networks: A review of methods and applications. *AI Open* **2020**, *1*, 57–81. [[CrossRef](#)]
24. Zhang, T.; Wang, X.; Xu, X.; Chen, C.P. GCB-Net: Graph convolutional broad network and its application in emotion recognition. *IEEE Trans. Affect. Comput.* **2019**, *13*, 379–388. [[CrossRef](#)]
25. Shen, L.; Sun, M.; Li, Q.; Li, B.; Pan, Z.; Lei, J. Multiscale Temporal Self-Attention and Dynamical Graph Convolution Hybrid Network for EEG-Based Stereogram Recognition. *IEEE Trans. Neural. Syst. Rehabil. Eng.* **2022**, *30*, 1191–1202. [[CrossRef](#)]
26. Song, T.; Zheng, W.; Song, P.; Cui, Z. EEG emotion recognition using dynamical graph convolutional neural networks. *IEEE Trans. Affect. Comput.* **2018**, *11*, 532–541. [[CrossRef](#)]
27. Li, Y. A Survey of EEG Analysis based on Graph Neural Network. In Proceedings of the 2021 2nd International Conference on Electronics, Communications and Information Technology (CECIT), Sanya, China, 27–29 December 2021; pp. 151–155.
28. Xu, K.; Hu, W.; Leskovec, J.; Jegelka, S. How powerful are graph neural networks? *arXiv* **2018**, arXiv:1810.00826.
29. Lun, X.; Yu, Z.; Chen, T.; Wang, F.; Hou, Y. A simplified CNN classification method for MI-EEG via the electrode pairs signals. *Front. Hum. Neurosci.* **2020**, *14*, 338. [[CrossRef](#)] [[PubMed](#)]
30. Acharya, U.R.; Oh, S.L.; Hagiwara, Y.; Tan, J.H.; Adeli, H. Deep convolutional neural network for the automated detection and diagnosis of seizure using EEG signals. *Comput. Biol. Med.* **2018**, *100*, 270–278. [[CrossRef](#)]
31. Pascanu, R.; Gulcehre, C.; Cho, K.; Bengio, Y. How to Construct Deep Recurrent Neural Networks. *arXiv* **2013**, arXiv:1312.6026.
32. Wolpaw, J.R.; Birbaumer, N.; Heetderks, W.J.; McFarland, D.J.; Peckham, P.H.; Schalk, G.; Donchin, E.; Quatrano, L.A.; Robinson, C.J.; Vaughan, T.M. Brain-computer interface technology: A review of the first international meeting. *IEEE Trans. Rehabil. Eng.* **2000**, *8*, 164–173. [[CrossRef](#)]
33. Rasheed, S.; Mumtaz, W. Classification of Hand-Grasp Movements of Stroke Patients using EEG Data. In Proceedings of the 2021 International Conference on Artificial Intelligence (ICAI), Islamabad, Pakistan, 5–7 April 2021; pp. 86–90.
34. Gers, F.A.; Schmidhuber, J.; Cummins, F. Learning to forget: Continual prediction with LSTM. *Neural Comput.* **2000**, *12*, 2451–2471. [[CrossRef](#)]
35. Huang, X.; Xu, Y.; Hua, J.; Yi, W.; Yin, H.; Hu, R.; Wang, S. A Review on Signal Processing Approaches to Reduce Calibration Time in EEG-Based Brain-Computer Interface. *Front. Neurosci.* **2021**, *15*, 1066. [[CrossRef](#)]
36. Khalaf, A.; Sejdic, E.; Akcakaya, M. Common spatial pattern and wavelet decomposition for motor imagery EEG-ftCD brain-computer interface. *J. Neurosci. Methods* **2019**, *320*, 98–106. [[CrossRef](#)]

37. Zeng, X.; Zhu, G.; Yue, L.; Zhang, M.; Xie, S. A feasibility study of ssvep-based passive training on an ankle rehabilitation robot. *J. Healthc. Eng.* **2017**, *2017*, 6819056. [[CrossRef](#)]
38. Majidov, I.; Whangbo, T. Efficient Classification of Motor Imagery Electroencephalography Signals Using Deep Learning Methods. *Sensors* **2019**, *19*, 1736. [[CrossRef](#)]
39. Levine, B.; Schweizer, T.; O'Connor, C.; Turner, G.; Gillingham, S.; Stuss, D.; Manly, T.; Robertson, I. Rehabilitation of Executive Functioning in Patients with Frontal Lobe Brain Damage with Goal Management Training. *Front. Hum. Neurosci.* **2011**, *5*, 9. [[CrossRef](#)] [[PubMed](#)]
40. Yu, S.; Zeng, Y.; Lei, W.; Wei, L.; Liu, Z.; Zhang, S.; Yang, J.; Wen, W. A Study of the Brain Abnormalities of Post-Stroke Depression in Frontal Lobe Lesion. *Sci. Rep.* **2017**, *7*, 13203.
41. Xu, R.; Jiang, N.; Vuckovic, A.; Hasan, M.; Mrachacz-Kersting, N.; Allan, D.; Fraser, M.; Nasserroleslami, B.; Conway, B.; Dremstrup, K. Movement-related cortical potentials in paraplegic patients: Abnormal patterns and considerations for BCI-rehabilitation. *Front. Neuroeng.* **2014**, *7*, 35. [[CrossRef](#)] [[PubMed](#)]
42. Feng, J.; Yin, E.; Jin, J.; Saab, R.; Daly, I.; Wang, X.; Hu, D.; Cichocki, A. Towards correlation-based time window selection method for motor imagery BCIs. *Neural. Netw.* **2018**, *102*, 87–95. [[CrossRef](#)] [[PubMed](#)]